International Journal of Grid Distribution Computing
Vol. 8, No.3, (2015), pp.11-22
http://dx.doi.org/10.14257/ijgdc.2015.8.3.02

Finding Useful Information for Big Data

Yong Shi

Department of Computer Science and Information Systems
Kennesaw State University
275 Kennesaw State University Rd NW
Kennesaw, GA 30144
yshi5@kennesaw.edu

Abstract

In this paper, we present our work on information analysis for big data. Big data is generated every day in various fields such as complex physics simulations, genomics, meteorology, as well as biological and environmental research. Traditional data mining applications cannot handle big data well because the data sets are so large and complex. In this paper we present our approach to analyzing the information that is hidden in the big data using various strategies. This proposed approach can assist to improve the performance of existing data analysis technologies, such as data mining approaches in Bioinformatics and other fields.

Keywords: *Big data, nearest neighboring search, clustering algorithms*

1. Introduction

With the advance of modern technology, big data is being generated at an explosive rate from various sources such as digital process, social media exchange, physics simulations, genomics, meteorology, and biological and environmental research [18, 23, 39, 40]. There are various fields of processing big data including analysis, capture, search, storage, visualization, etc. Traditional approaches fail to capture, manage, and process data within a limited time. Advanced technology and approaches are needed to extract meaningful information and value from big data.

Due to the size of the big data, it is not easy to find useful and valuable information in big data. Patterns are normally hidden in the high dimensionality and large size of big data. How to find interesting patterns in big data is one of the big challenges in the big data research field. In order to improve the pattern recognition approaches, we need to analyze the similarity between data subgroups and similarity between individual data points. In traditional nearest neighbor problems, the similarity between two data points is based on a similarity function which aggregates the difference between each dimension of the two data points. However, such approaches only focus on full similarities, without considering patterns hidden in subspaces [1, 12, 14, 54]. Although there are some approaches designed for subspace pattern similarity recognition [5, 6, 30], they require extra information such as the dimensionality of subspaces.

Pattern recognition is a branch of machine learning which is studied in various research fields such as computer science, psychology, psychiatry, ethology and cognitive science. It focuses on finding useful and interesting patterns in big data sets. Pattern recognition normally applies supervised learning which uses labeled training data for the assignment of a label to a given input value. One of the well-known examples of pattern recognition is classification. Classification focuses on assigning each input value to one of a given set of classes [34, 51, 17, 15, 22].

The remainder of this paper is organized as follows. Section 2 introduces the related research work in data mining fields. Section 3 presents the formalization and definitions

ISSN: 2005-4262 IJGDC

of the problem, and describes the pattern searching algorithm for big data. Section 4 presents experimental results, and concluding remarks are offered in Section 5.

2. Related Work

There are many research fields in data mining that are related to pattern searching in big data [56, 57, 43, 4, 52]. One of them is cluster analysis. Cluster analysis is applied to identify homogeneous and well-separated groups of objects in data sets, which plays an important role in fields of business and science. There are various steps such as data cleaning, feature selection, application of a clustering algorithm, validation of results, and interpretation of the results [26]. Among these steps, the clustering algorithm and validation of the results are especially critical. There are different types of clustering algorithms [33] such as: partitioning [32, 37, 42, 19], hierarchical [13, 28, 29], grid-based [53, 48, 9], and density-based [24, 31, 10] algorithms.

Partitioning algorithms request an input parameter which is the number of clusters K. It constructs a partition of a data set into a set of K clusters. Partitioning algorithms start with an initial partition. It then uses an iterative control strategy to optimize the quality of the clustering results by moving objects from one group to another. Hierarchical algorithms represent the structure of a given data set in a hierarchical way. The hierarchical decomposition is represented by a tree structure which is normally called dendrogram. Grid-based algorithms do not process a given data set directly. Instead, they divide the data space into several grids and perform all operations on the grids. Thus grid-based algorithms process the data set fast no matter how large a given data set is. The processing time is dependent only on the number of segments of each dimension in the quantized space. Density-based algorithms are designed based on the definition of density in a given data set. They are based on the criteria that for each point within a cluster, the density in the neighborhood of a given radius must exceed a defined threshold. Density-based algorithms can discover clusters of arbitrary shapes, and they can also filter out outliers. Each existing clustering algorithm has its advantages and disadvantages. None of the algorithms can detect clusters from all kinds of data sets efficiently and effectively. For a data set with clusters of various sizes, density, or shape, different clustering algorithms are best suited to detecting clusters of different types in the data set.

Outlier detection is closely related to cluster analysis. An outlier is a data point that does not follow the main characteristics of the input data. Outlier detection is concerned with discovering the exceptional behaviors of certain objects, and it is useful in numerous applications, including credit card fraud detection, discovery of criminal activities, discovery of computer intrusion, etc. Various research works have been done on outlier detection [55, 50, 59, 44]. Yu, D. *et. al.* [61] proposed an outlier detection approach termed FindOut as a byproduct of WaveCluster [48]. In their approach, clusters are removed from the original data, and outliers are identified. Knorr, E. M. *et. al.* [38] detected a distance-based outlier defined as a data point, if this data point has a certain percentage of the data points in the same data set which have distances of more than a predefined value away from it. Ramaswamy, S. *et. al.* [45] further extended Knorr's work based on the distance of a data point from its *k*th nearest neighbor and identified the top n points with largest kth nearest neighbor distances as outliers. Breunig, M. M. *et. al.* [16] presented the concept of local outlier. They also defined local outlier factor of a data point which is a degree of how isolated the data point is from the surrounding neighborhood. Aggarwal *et. al.* [8] extended the problem of outlier detection in subspace, and one of its purposes is to overcome the curse of dimensionality. Shi, Y. *et. al.* [49] tried to detect and adjust the set of clusters and outliers according to both the intra-relationship and the inter-relationship between the set of clusters and the set of outliers. The authors discussed the issue related to subspace-based approach, in which each cluster is associated with a unique subset of dimensions, so is each outlier. In each iteration, the subset of dimensions

and the quality of each cluster are modified and adjusted dynamically, so are the subset of dimensions and the quality of each outlier.

There are several research works on determining the quality of acquired clusters [20, 41, 58]. Ng, R. T. *et. al.* [29] compared the aggregated inter-connectivity of two clusters to measure their similarity. Karypis, G. *et. al.* [36] measured the similarity of two clusters based on a dynamic model in which two clusters are merged only if the inter-connectivity and similarity between them are highly related to the internal interconnectivity of the clusters and closeness of items within the clusters. Halkidi, M. *et. al.* [41] defined a validity index to describe the information of the average degree of scatter within clusters and the average number of points between the clusters. The index also incorporated criteria addressing compactness, separation and density. Chen, C. F. *et. al.* [20] introduced a fuzzy validity function and they measured the overall average compactness and separation of the fuzzy partition using the function. For example, the average compactness was estimated by the deviation of data points from the center of each cluster, and the separation of the partition was represented by the distance between cluster centers. All these clustering validity measurements evaluated clustering algorithms by measuring the overall quality of the clusters.

The similarity search problem has been studied and many algorithms haves been proposed to solve the K nearest neighbor search [27, 47, 2, 3, 25, 21]. In traditional nearest neighbor problems, we use a similarity function such as Euclidean distance to measure the similarity between two data points, which aggregates the difference between two data points on each dimension. Normally the nearest neighbor problems are solved based on the distance between the data point and the query point over a fixed set of dimensions. However, such approaches only focus on the similarity in full data space of the data set. Traditional algorithms [1, 12, 54] suffer from the "cure of dimensionality" which makes the distance between two data points in high dimensionality less meaningful. In a high dimensional space the data are usually sparse. The traditional similarity function such as Euclidean distance may not work well as dimensionality goes higher. Some research [14] shows that in high dimensions nearest neighbor queries become unstable because of that. Some approaches [30, 6, 5] are proposed for partial similarities. However, they have limitations. For example, they require the fixed subset of dimensions or fixed number of dimensions as the input parameters for the algorithms.

High dimensional data sets continue to pose a challenge to data mining algorithms at a very fundamental level. It is well acknowledged that in the real world a large proportion of data has irrelevant features and those features may cause a reduction in the accuracy of some algorithms. Dimension reduction is one of the well-known techniques for improving the data analysis performance [9, 7, 46]. In dimension reduction approaches, a data set is transformed to a lower dimensional space, but it still preserves the major information it carries, so that further processing can be simplified without compromising the quality of the final results. Dimension reduction is often used in clustering, classification, and many other machine learning and data mining applications. There are various solutions to reduce dimensions. One approach is to identify important attributes based on input from domain experts, selecting a subset of attributes of existing dimensions. Another type of approaches are projection methods in which the new projected dimensions are linear or un-linear combination of old dimensions. For example, some approaches use principal component analysis through singular value decomposition [35] for numerical attributes and they define new attributes (principal components) as mutually-orthogonal linear combinations of the original attributes. In information retrieval, latent semantic indexing uses singular value decomposition to project textual documents represented as document vectors. Singular value decomposition was shown to be the optimal solution for a probabilistic model for document/word occurrence. However, these approaches have a major drawback in that the generated low dimensional subspace has no intuitive meaning to users. Some approaches use random projections to subspaces as well.

Data preprocessing procedures can greatly benefit the utilization and exploration of real data. For example, Shrinking [60] is a data preprocessing technique which optimizes the inner structure of data. This technique can be applied in many data mining fields.

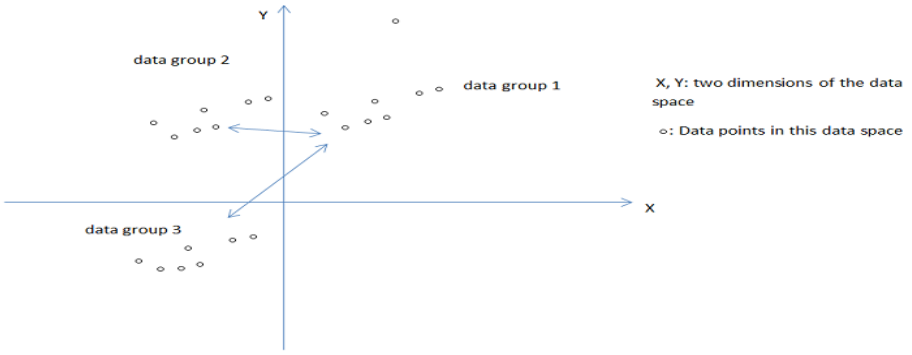

Figure 1. An Example of Three Similar Data Groups in a Two Dimensional Data Space

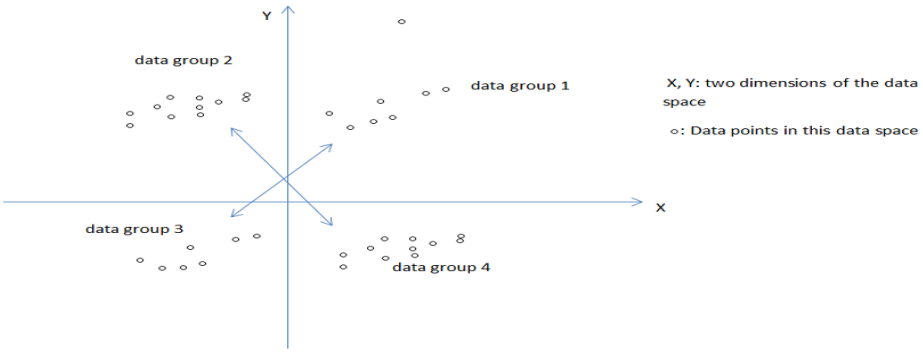

Figure 2. An Example of Two Sets of Similar Data Groups in a Two Dimensional Data Space

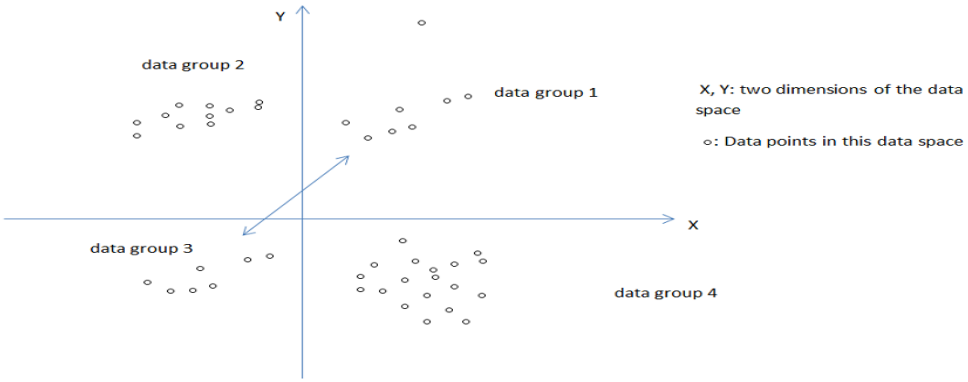

Figure 3. An Example of Three Types of Data Groups in a Two Dimensional Data Space

3. Finding Patterns for Big Data

Patterns exist in various data space of big data. Figure 1-5 demonstrates five examples of patterns. Here we use two dimensional data space just for demonstration; however, the dimensionality of big data in the real world applications is much higher. Figure 1 shows a two dimensional data set that contains three data groups: data group 1- 3. Data group 2 and 3 and both similar to data group 1. Thus, these three data groups have the same pattern. Figure 2 shows a two dimensional data set that contains four data groups: data group 1- 3 and 4. Data group 1 is similar to data group 3, and data group 2 is similar to data group 4. Thus, there are two patterns in this data set. Figure 3 also shows a two dimensional data set that contains four data groups: data group 1- 3 and 4. Data group 1 is similar to data group 3, but neither data group 2 nor data group 4 is similar to any other data groups. Figure 4 shows a two dimensional data set that contains four data groups: data group 1- 3 and 4. In this data set, there are no data groups that are similar to each other. Figure 5 shows a two dimensional data set that contains four data groups: data group 1- 3 and 4. In this data set, data groups 2 and 3 are across the x axis and y axis.

Figure 4. An Example of Four Types of Data Groups in a Two Dimensional Data Space

Figure 5. An Example of Four Types of Data Groups Across X axis and Y axis in a Two Dimensional Data Space

From the examples of data sets and patters shown in figure 1, 2, 3, 4 and 5, we can see that there are various conditions how patterns of data groups exist in data sets depending on the real world condition of the data sets. In this section we propose to design an algorithm which analyzes the similarity of data points and detect patterns in big data.

Suppose we denote n as the total number of data points and d be the dimensionality of the data space. Let Dl be the *l*th dimension, where l = 1, 2, ..., d. Let the input d-

dimensional data set be DS = {X1, X2… Xn}, which is normalized to be within the hypercube [0; 1]d⊂Rd. Each data point Xi is a d-dimensional vector [*xi1, xi2... xid*].

Definition 1: Dimensional Related: Given two data points Xi ∈ DS and Xj ∈DS, if |Xil –Xjl| < δ, where l = 1, 2... d, we call Xi and Xj are dimensional related on Dl:

$$Xi \ DR \ Xj \quad (1)$$

Definition 2: Pattern Related: Given two data points Xi ∈ DS and Xj ∈ DS, if Xi DR Xj on more than p percentage of the d dimensions, we call Xi and Xj are pattern related:

$$Xi \ PR \ Xj \quad (2)$$

Definition 3: Pattern: Suppose there is a set E of data points ∈ DS. If for any Xk ∈ E, there is Xm ∈ E, so that Xk PR Xm, we call E a pattern in DS.

Definition 4: Maximum Pattern: Suppose ME is a pattern in DS. If for any Xk ∈ ME, there is no Xm ∉ ME, so that Xk PR Xm, we call ME a maximum pattern in DS.

Given a data set DS of n data points X = {X1, X2, …, Xn} with d dimensions Dl, D2, …, Dd, we first sort the data points on each dimension Dl, l=1, 2, ..., d. This is to find groups of data points on each dimension which are close to each other. We locate each pair of data points that are dimensional related to each other on each dimension.

Next, we aggregate the result from all the dimensions, and find each pair of data points that are pattern related to each other. Then, we find patterns based on the pattern related relationship between data points. There are various pattern generated, however, not all of them are maximum patterns. The next step is to merge patterns to form maximum pattern when necessary, to make sure there are no pair of pattern related data points that are not in the same pattern. Figure 6 presents our algorithm.

Algorithm Pattern Search

Purpose: given a data set DS, analyze the hidden information in DS

Input: DS: data set, D: dimensions, ϵ: dimensional related threshold, p: pattern related threshold.

Output: patterns in DS

Begin

1) Sort the data points on each dimension D_l, l=1, 2, ..., d;
2) For each dimension D_l, find pairs of data points X_i and X_j so that X_i DR X_j;
3) In the d-dimensional data space, find pairs of data points X_i and X_j so that X_i PR X_j;
4) Find the maximum patterns in DS;
5) Output the result.

End.

Figure 6. Algorithm Pattern Search

4. Experiments

We perform experiments on both synthetic and real data sets to assess the accuracy and efficiency of the proposed approach, which are run on Intel(R) Pentium(R) 4 with CPU of 3.39GHz and Ram of 0.99 GB. We first generate synthetic data sets to test the scalability

of our algorithm over dimensionality and data size. The sizes of the data sets vary from 10,000, 15,000, ... to 200,000, with the gap of 5,000 between each two adjacent data set sizes, and the dimensions of the data sets vary from 5, 10 ... to 100, with the gap of 5 between each two adjacent numbers of dimensions. The experiments on synthetic data sets demonstrate that our algorithm is scalable over dimensionality and data size.

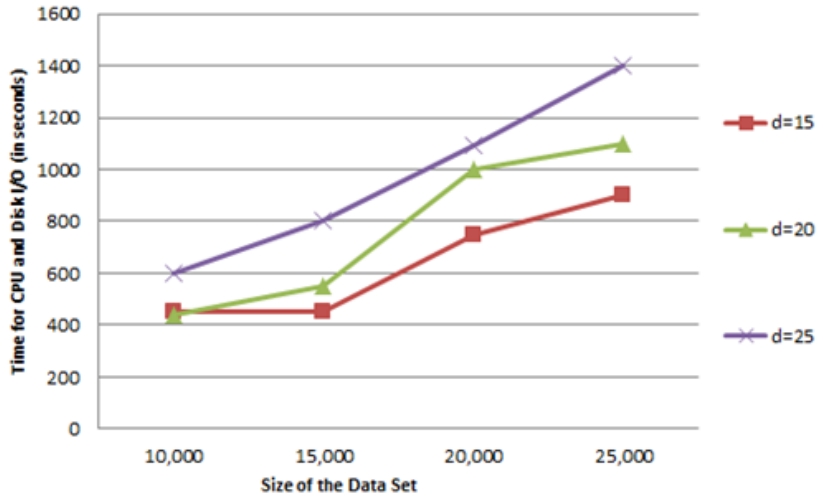

Figure 7. Running Time with Increasing Data Size

Figure 7 shows the running time of various data sets with data size increasing from 10,000 to 25,000. The data sets have different dimensions such as 15, 20 and 25.

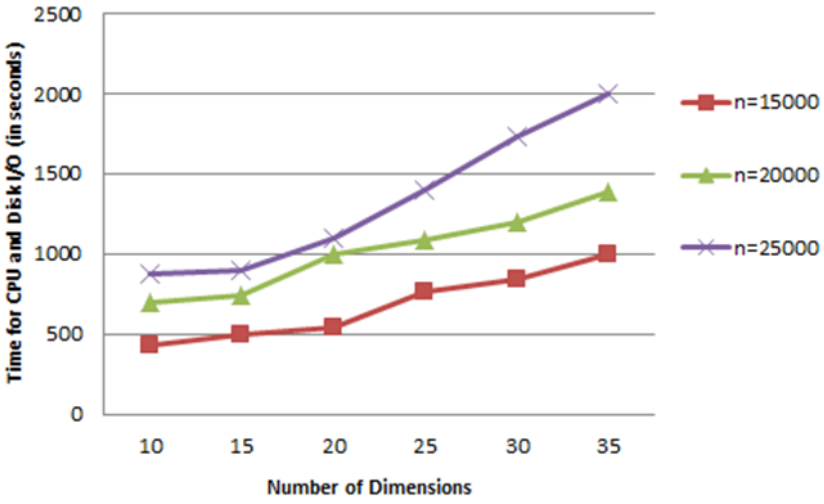

Figure 8. Running Time with Increasing Dimensionality

Figure 8 shows the running time of various data sets with dimensionality increasing from 10 to 35. The data sets have different data sizes such as 15000, 20000 and 25000.

From Figure 7 and 8 we can see that our algorithm is scalable when data size and dimensionality go higher.

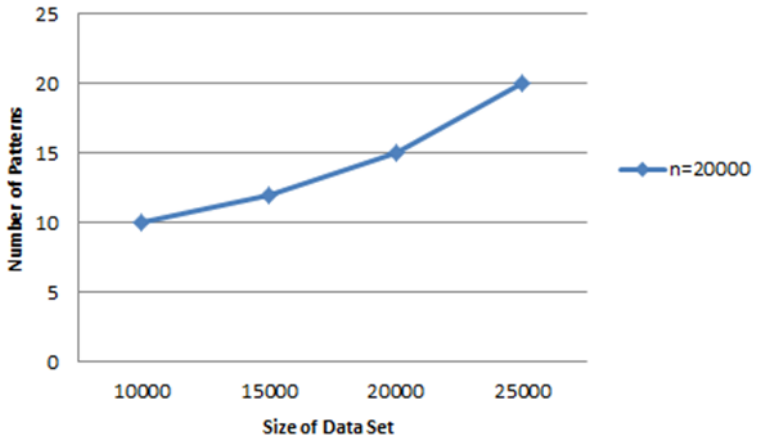

Figure 9. Number of Patterns with Increasing Data Size

Figure 9 shows the change of number of patterns when data size increases from 10,000 to 25,000.

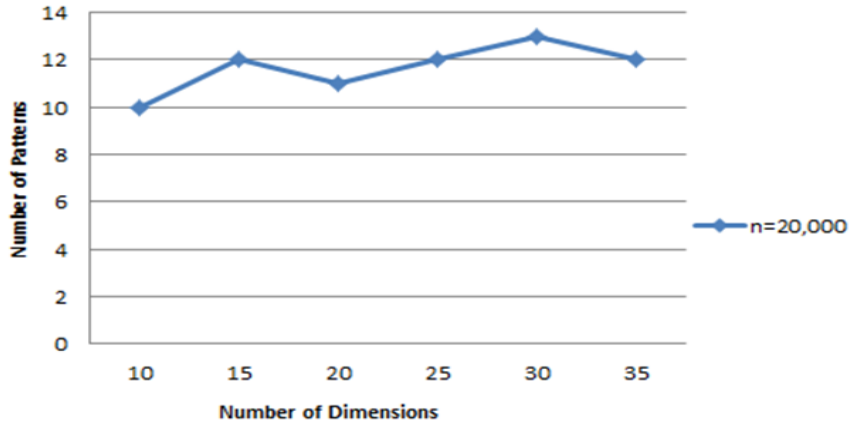

Figure 10. Number of Patterns with Increasing Dimensionality

Figure 10 shows the change of number of patterns when the dimensionality increases from 10 to 35.

From Figure 9 and 10 we can see that the number of patterns does not have significant change when the data size or the dimensionality goes higher.

We also use real data sets from UCI Machine Learning Repository [11] to demonstrate the effectiveness of our algorithm. One of the real data sets we use is called Wine Recognition data set, which contains the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. There are 178 instances, each of which has 13 dimensions including alcohol, magnesium, color intensity, etc. Three natural clusters are in the data set, with various sizes of 59, 71 and 48. We perform the algorithms on the Wine Recognition data set, with the accuracy rate of 93.75%.

5. Conclusion

Big data is generated nowadays from many different research and industrial fields. In this paper we present our strategy to find useful patterns in big data. We analyze the distribution of the data set, and perform dimension to dimension process to find useful

information. Our approach can be applied in many fields such as bioinformatics, pattern recognition, data clustering and signal processing.

Acknowledgement

This paper is a revised and expanded version of a paper entitled "Towards Information Analysis for Big Data" presented at the 7th International Conference on Control and Automation, Mingguang Haikou, Hainan, China, December 20-13, 2014.

References

[1] D. A. White and R. Jain, "Similarity Indexing with the SS-tree", In Proceedings of the 12th Intl. Conf. on Data Engineering, New Orleans, Louisiana, February **(1996)**, pp.516–523.
[2] E. Achtert, C. B¨ohm, P. Kr¨oger, P. Kunath, A. Pryakhin, and M. Renz, "Efficient reverse k-nearest neighbor search in arbitrary metric spaces", In SIGMOD '06, , New York, NY, USA, **(2006)**, pp. 515–526, ACM.
[3] E. Achtert, C. B¨ohm, P. Kr¨oger, P. Kunath, A. Pryakhin, and M. Renz, "Efficient reverse k-nearest neighbor estimation", In BTW, **(2007)**, pp. 344–363.
[4] E. Achtert, H. P. Kriegel, and A. Zimek. Elki, "A software system for evaluation of subspace clustering algorithms", In SSDBM, **(2008)**, pp. 580–585.
[5] C. C. Aggarwal, "Towards meaningful high-dimensional nearest neighbor search by human computer interaction", In ICDE, **(2002)**.
[6] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space", Lecture Notes in Computer Science, **(1973)**, **(2001)**.
[7] C. C. Aggarwal, C. Procopiuc, J. Wolf, P. Yu, and J. Park, "Fast algorithms for projected clustering", In Proceedings of the ACM SIGMOD CONFERENCE on Management of Data, pp. 61–72, Philadelphia, PA, **(1999)**.
[8] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data", In SIGMOD Conference, **(2001)**.
[9] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications", In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 94–105, Seattle, WA, **(1998)**.
[10] M. Ankerst, M. M. Breunig, H. P. Kriegel and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure", Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD' 99), Philadelphia, PA, **(1999)**, pp. 49–60.
[11] S. D. Bay, "The UCI KDD Archive [http://kdd.ics.uci.edu]", University of California, Irvine, Department of Information and Computer Science.
[12] D. A. Berchtold, S. Keim and H. P. Kriegel, "The X-tree: An index structure for high dimensional data", In VLDB'96, Bombay, India, **(1996)**, pp. 28–39.
[13] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, , Montreal, Canada, **(1996)**, pp. 103–114.
[14] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?", In International Conference on Database Theory 99, Jerusalem, Israel, **(1999)**, pp. 217–235.
[15] J. C. Bezdek, M. R. Pal, J. Keller, and R. Krisnapuram, "Fuzzy Models and Algorithms for Pattern Recognition and Image Processing", Kluwer Academic Publishers, Norwell, MA, USA, **(1999)**.
[16] M. Breunig, H. Kriegel, R. Ng, and J. Sander, "LOF: Identifying density-based local outliers", In Proceedings of the ACM SIGMOD CONFERENCE on Management of Data, , Dallas, Texas, May 16-18 **(2000)**, pp. 93– 104
[17] C. J. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, vol. 2, **(1998)**, pp. 121–167.
[18] S. Chan, W. Rhodes, C. Atencio, C. Kuo, B. Ranalli, A. Miao, S. Sala, S. Serene, R. Helbling, S. Rumbley, M. Clement, L. Sokol, and L. Gary, "Robust decision engineering", Collaborative big data and its application to international development/aid. IEEE, vol. 12, **(2012)**.
[19] Y. Chen and L. Tu, "Density-based clustering for real-time stream data", In KDD, **(2007)**, pp. 133–142.
[20] C. Chen and J. Lee, "The Validity Measurement of Fuzzy C-Means Classifier for Remotely Sensed Images", In Proc. ACRS 2001 - 22nd Asian Conference on Remote Sensing, **(2001)**.
[21] B. Cui, H. Shen, J. Shen, and K. Tan, "Exploring bit-difference for approximate KNN search in high-dimensional databases", In Australasian Database Conference, **(2005)**.
[22] R. L. Das, B. K. Prasad, and G. Sanyal, "Article: Hmm based offline handwritten writer independent English character recognition using global and local feature extraction", International Journal of Computer Applications, vol. 46, no. 10, **(2012)**, pp. 45–50, May 2012. Full text available.

[23] B. Elser and A. Montresor, "An evaluation study of big data frameworks for graph processin", In Big Data, 2013 IEEE International Conference on, **(2013)** October, pp. 60–67.
[24] M. Ester, K. H. P., J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, **(1996)**.
[25] R. Fagin, R. Kumar, and D. Sivakumar, "Efficient similarity search and classification via rank aggregation", **(2003)**.
[26] U. M. Fayyad, G. P. Shapiro, P. Smyth, and R. Uthurusamy, "Advances in Knowledge Discovery and Data Mining", AAAI Press, **(1996)**.
[27] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing", In The VLDB Journal, **(1999)**, pp. 518–529.
[28] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases", In Proceedings of the ACM SIGMOD conference on Management of Data, **(1998)**, pp. 73–84, Seattle, WA.
[29] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes", In Proceedings of the IEEE Conference on Data Engineering, **(1999)**.
[30] A. Hinneburg, C. C. Aggarwal, and D. A. Keim, "What is the nearest neighbor in high dimensional spaces?" In The VLDB Journal, **(2000)**, pp. 506–515.
[31] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise", In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, **(1998)** August, pp. 58–65, New York.
[32] J. Mac Queen, "Some methods for classification and analysis of multivariate observations", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Statistics, **(1967)**.
[33] A. Jain, M. Murty and P. Flyn, "Data clustering: A review", ACM Computing Surveys, vol. 31, no. 3, **(1999)**.
[34] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 22, no. 1, **(2000)**, pp. 4–37.
[35] K. R. Kanth, D. Agrawal, and A. Singh, "Dimensionality reduction for similarity searching in dynamic databases", In Proceedings of the ACM SIGMOD CONFERENCE on Management of Data, pp.166–176, Seattle, WA, **(1998)**.
[36] G. Karypis, E.-H. S. Han and V. K. News, "Chameleon: Hierarchical clustering using dynamic modeling", Computer, vol. 32, no. 8, **(1999)**, pp. 68–75.
[37] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: an Introduction to Cluster Analysis" John Wiley & Sons, **(1990)**.
[38] E. Knorr and R. Ng, "Algorithms for mining distance-based outliers in large datasets", In Proceedings of the 24th VLDB conference, **(1998)**, pp. 392–403, New York, August.
[39] D. Kumar, M. Palaniswami, S. Rajasegarar, C. Leckie, J. C. Bezdek and T. C. Havens, "clusivat: A mixed visual/numerical clustering algorithm for big data", In Proceedings of the 2013 IEEE International Conference on Big Data, Santa Clara, CA, USA, pp. 112–117, **(2013)** October 6-9.
[40] A. Madkour, W. G. Aref and S. Basalamah, "Knowledge cubes - A proposal for scalable and semantically-guided management of big data", In Proceedings of the 2013 IEEE International Conference on Big Data, Santa Clara, CA, USA, **(2013)** October 6-9, pp. 1–7.
[41] M. Halkidi and M. Vazirgiannis, "A Data Set Oriented Approach for Clustering Algorithm Selection", In PKDD, **(2001)**.
[42] R. T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining", In Proceedings of the 20th VLDB Conference, **(1994)**, pp. 144–155, Santiago, Chile,.
[43] M. Q. Nguyen, L. Mark, and E. Omiecinski, "Unusual pattern detection in high dimensions", In PAKDD, **(2008)**, pp. 247–259,.
[44] G. L. Peterson and B. T. McBride, "The importance of generalizability for anomaly detection", Knowl. Inf. Syst., vol. 14, no. 13, **(2008)**, pp. 377–392.
[45] S. Ramaswamy, R. Rastogi and K. Shim, "Efficient algorithms for mining outliers from large data sets", In Proceedings of the ACM SIGMOD CONFERENCE on Management of Data, **(2000)** May 16-18, pp. 427–438, Dallas, Texas.
[46] T. Seidl and H. Kriegel, "Optimal multi-step k-nearest neighbor search", In Proceedings of the ACM SIGMOD conference on Management of Data, **(1998)**, pp. 154–164, Seattle, WA,.
[47] T. Seidl and H.-P. Kriegel, "Optimal multi-step k-nearest neighbor search", SIGMOD Rec., vol. 27, no. 2, **(1998)**, pp.154–165.
[48] G. Sheikholeslami, S. Chatterjee and A. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases", In Proceedings of the 24th International Conference on Very Large Data Bases, **(1998)**.
[49] Y. Shi, Subcoid, "Exploring cluster-outlier iterative detection approach to multi-dimensional data analysis in subspace", In ACMSE 2008: The 46th ACM Southeast Conference, Auburn, AL, USA, **(2008)**.
[50] Y. Tao, X. Xiao, and S. Zhou, "Mining distance-based outliers from large databases in any metric space", In KDD, **(2006)**, pp. 394–403.

[51] A. Vailaya, A. Jain and H. J. Zhang, "On image classification: City images vs. landscapes", PATTERN RECOGNITION, vol. 31, **(1998)**, pp. 1921–1935.
[52] J.-S. Wang and J.-C. Chiang, "A cluster validity measure with outlier detection for support vector clustering", IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 38, no. 1, **(2008)**, pp. 78–89.
[53] W. Wang, J. Yang and R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining", In Proceedings of the 23rd VLDB Conference, **(1997)**, pp. 186–195, Athens, Greece.
[54] R. Weber, H.-J. Schek and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces", In Proc. 24th Int. Conf. Very Large Data Bases, VLDB, vol. 24, no. 27, **(1998)**, pp. 194–205.
[55] M. Wu and C. Jermaine, "Outlier detection by sampling with accuracy guarantees", In KDD, pp. 767–772, **(2006)**.
[56] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand and D. Steinberg, "Top 10 algorithms in data mining", Knowl. Inf. Syst., vol. 14, no. 1, **(2008)**, pp. 1–37.
[57] H. Xiong, M. Steinbach, A. Ruslim and V. Kumar, "Characterizing pattern preserving clustering", Knowl. Inf. Syst., vol. 19, no. 3, **(2009)**, pp. 311–336.
[58] H. Xiong, J. Wu and J. Chen, "K-means clustering versus validation measures: a data distribution perspective", In KDD, **(2006)**, pp. 779–784.
[59] J. Yang, N. Zhong, Y. Yao, and J.Wang, "Local peculiarity factor and its application in outlier detection", In KDD, **(2008)** pp. 776–784.
[60] Y. Shi, Y. Song and A. Zhang, "A shrinking-based approach for multidimensional data analysis", in the 29th VLDB conference, **(2003)** September.
[61] D. Yu, G. Sheikholeslami and A. Zhang, "Find out: Finding outliers in very large datasets", The Knowledge and Information Systems (KAIS), vol. 4, **(2000)** October.

Author

Yong Shi, received the BS and MS degrees, both in computer science, from the University of Science and Technology of China in 1996 and 1999, respectively. He received Ph.D. in computer science from the State University of New York at Buffalo in 2006. He is currently an associate professor in the Department of Computer Science in Kennesaw State University. His research interests include data mining, database, machine learning, and information retrieval.